09/718,679

AMENDMENTS TOTHE CLAIMS:

1-7. (Canceled)

8. (Currently Amended) A method for improving TCP throughput over lossy communication links without affecting performance over non-lossy links comprising:

[[-]] determining lookahead-loss which is the number of lost packets in a given loss-window~~,~~;

[[-]] using said loss-window and said lookahead loss to detect congestion in said communication links; ~~and~~

[[-]] ~~controlling transmission under congestion conditions as well as under normal conditions,~~ choosing a first transmission protocol if congestion is below a first threshold;

choosing a second transmission protocol if congestion is between said first threshold and a second threshold;

choosing a third transmission protocol if congestion is between said second threshold and a third threshold; and

returning to a previous protocol when packet loss exceeds a predetermined limit.

9. (Currently Amended) The method as claimed in claims 8, wherein said determining of lookahead-loss is for identifying the number of packets transmitted by ~~the~~ a sender in said loss-window for which ~~either~~ at least one of the following conditions ~~are~~ is true:

[[-]] said sender has received at least max-dupacks ~~(an appropriately selected number, typically three)~~ duplicate cumulative acknowledgements, and

[[-]] said sender has neither received acknowledgement nor selective acknowledgment for said packets, while it has received selective acknowledgements for at least max-dupsacks ~~(an appropriately selected number, typically three)~~ packets with higher sequence numbers.

10. (Original) The method as claimed in claim 8, wherein said detecting of congestion is for identifying when the number of packets lost in a loss-window is greater than an appropriately selected preset number.

11. (Currently Amended) The method as claimed in claim 8, wherein said ~~controlling~~ first transmission protocol is a TCP k-SACK protocol which is a modification of a fast retransmit algorithm of a basic congestion control algorithm of TCP to include [-] entering a 'halt growth phase' whenever said lookahead loss is greater than zero congestion is not detected, and [-] entering a k-recovery phase' whenever the congestion is detected.

12. (Currently Amended) The method as claimed in claim 11, wherein during said 'halt growth phase'~~, the~~ a sender freezes ~~the~~ a congestion window and maintains it in that state.

13. (Original) The method as claimed in claim 11, wherein said entry into 'k-recovery phase' reduces a congestion window to half its original size, while a slow-start threshold is reduced to half only on a first occasion of entry into the k-recovery phase during a packet loss recovery cycle.

14. (Currently Amended) The method as claimed in claim 11, ~~farther~~ further including:

[[-]] entering a "Post Recovery" phase wherein the sender continues in congestion avoidance or slow-start phase at the end of the ~~fast~~ recovery phase,

~~more~~ and provides an accurate estimation of pipe size using ~~the~~ a received selective acknowledgement (SACK) data, and

[[-]] use of said accurate pipe size information for controlling window inflation and deflation thereby allowing ~~quicker~~ retransmission of lost packets and resulting ~~faster~~ recovery.

15. (Currently Amended) A computer program product comprising computer readable program code stored on computer readable storage medium embodied therein for improving TCP throughput over lossy communication links without affecting performance over non-lossy links comprising:

[[-]] a computer readable program code ~~means~~ configured for determining lookahead-loss which is the number of lost packets in a given loss-window,

[[-]] wherein said computer readable program code ~~means~~ configured for using said loss-window and said lookahead loss to detect congestion in said communication links, ~~and~~

[[-]] wherein said computer readable program code ~~means~~ configured for ~~controlling transmission under congestion conditions as well as under normal conditions~~ choosing a first transmission protocol if congestion is below a first threshold,

wherein said computer readable program code configured for choosing a second transmission protocol if congestion is between said first threshold and a second threshold,

wherein said computer readable program code configured for choosing a third transmission protocol if congestion is between said second threshold and a third threshold, and

wherein said computer readable program code configured for returning to a previous protocol when packet loss exceeds a predetermined limit.

16. (Currently Amended) The computer program product as claimed in claim 15, wherein said computer readable program code means configured for determining lookahead-loss is a mechanism for identifying the number of packets transmitted by the sender in said loss-window, for which ~~either~~ at least one of the following conditions is true:

[[-]] said sender has received at least max-dupacks ~~(an appropriately selected number, typically three)~~ duplicate cumulative acknowledgements,

[[-]] said sender has neither received acknowledgement nor selective acknowledgement for said packets, while it has received selective acknowledgements for at least max-~~dupsacks (an appropriately selected number, typically three)~~ packets with higher sequence numbers.

17. (Original) The computer program product as claimed in claim 15, wherein said computer readable program code means configured for detecting congestion is a mechanism for identifying when the number of packets lost in a loss-window is greater than an appropriately selected preset number.

18. (Currently Amended) The computer program product as claimed in claim 15, wherein said computer readable program code configured for ~~controlling~~ choosing said first transmission protocol is a TCP k-SACK protocol which is a modification of a fast retransmit algorithm of a basic congestion control algorithm of TCP to include [[-]] entering a 'halt growth phase' whenever said lookahead loss is greater than zero and congestion is not detected, and [[-]] entering a k-recovery phase' whenever the congestion is detected.

19. (Currently Amended) The computer program product as claimed in claim 18, wherein during said 'halt growth phase'~~, the~~ a sender freezes ~~the~~ a congestion window and maintains it in that state.

20. (Currently Amended) The computer program product as claimed in claim 18, wherein said entry into 'k-recovery phase' reduces a congestion window to half its original size, while a slow-start threshold is reduced to half only on a first occasion of entry into the k-recovery phase during a packet loss recovery cycle.

21. (Currently Amended) The computer program product as claimed in claim 18, ~~thither~~ further including:

[[-]] entering a "Post Recovery" phase wherein the sender continues in congestion avoidance or slow-start phase at the end of the ~~fast~~ recovery phase,

~~more~~ and provides an accurate estimation of pipe size using ~~the~~ a received selective acknowledgement (SACK) data, and

[[-]] use of said accurate pipe size information for controlling window inflation and deflation thereby allowing ~~quicker~~ retransmission of lost packets and resulting ~~faster~~ recovery.

22. (New) A method for improving TCP throughput over lossy communication links without affecting performance over non-lossy links comprising:

determining lookahead-loss which is the number of lost packets in a given loss-window;

using said loss-window and said lookahead loss to detect congestion in said communication links; and

controlling transmission under congestion conditions as well as under normal conditions,

wherein said controlling transmission comprises controlling a size of said loss-window by:

beginning in a slow-start phase;

advancing to a congestion avoidance phase when a slow-start threshold is reached;

entering a halt growth phase when a first level of packet loss occurs;

returning to said congestion avoidance phase when a first level of packet recovery occurs;

entering a k-recovery phase when a second level of packet loss occurs,

wherein said loss window and said slow-start threshold are reduced in half and said congestion avoidance phase is restarted.

23. (New) The method as claimed in claim 22, wherein said determining of lookahead-loss is for identifying the number of packets transmitted by a sender in said toss-window for which at least one of the following conditions is true:

said sender has received at least max-dupacks duplicate cumulative acknowledgements, and said sender has neither received acknowledgement nor selective acknowledgment for said packets, while it has received selective acknowledgements for at least mnax-dupsacks packets with higher sequence numbers.

24. (New) The method as claimed in claim 22, wherein said detecting of congestion is for identifying when the number of packets lost in said loss-window is greater than an appropriately selected preset number.

25. (New) The method as claimed in claim 22, wherein said controlling transmission is a TCP k-SACK protocol which is a modification of a fast retransmit algorithm of a basic congestion control algorithm of TCP to include entering a 'halt growth phase' whenever said lookahead loss is greater than zero congestion is not detected, and entering 'a k-recovery phase' whenever the congestion is detected.

26. (New) The method as claimed in claim 25, wherein during said 'halt growth phase' a sender freezes a congestion window and maintains it in that state.

27. (New) The method as claimed in claim 25, wherein said entry into 'k-recovery phase' reduces a congestion window to half its original size, while a slow-start threshold is reduced to half only on a first occasion of entry into the k-recovery phase during a packet loss recovery cycle.

28. (New) The method as claimed in claim 25, further including:

entering a "Post Recovery" phase wherein the sender continues in congestion avoidance or slow-start phase at the end of the recovery phase, and provides an accurate estimation of pipe size using a received selective acknowledgement (SACK) data, and use of said accurate pipe size information for controlling window inflation and deflation thereby allowing retransmission of lost packets and resulting recovery.